Problem Solving

- ✓ Problem solving, particularly in artificial intelligence, may be characterized as a systematic search through a range of possible actions in order to reach some predefined goal or solution.

Problem Solving

- ✓ Problem Solving is fundamental to many AI-based applications.
- ✓ There are two types of problems:
  - The problem like computation of sine of a particular angle or a square root of a number. We can solve these kind of problems through deterministic procedure.
  - In the real world, very few problems lend themselves to straightforward solutions.
- ✓ Many real world problems can be solved only by searching for a solution.
- ✓ AI is concerned with these kind of problem solving.

State

- ✓ A state is a representation of elements at a given moment.
- ✓ A problem is defined by its elements and their relations.
- ✓ At each instant of a problem, the elements have specific descriptors and relations; the descriptors tell - how to select elements ?
- ✓ Among all possible states, there are two special states called

  - Initial State is the start point.

  - Final State is the goal state.

State

- ✓ A Successor Function is needed for state change.
- ✓ The successor function moves one state to another state.
- ✓ It
  - – is a description of possible actions; a set of operators.
  - – is a transformation function on a state representation, which converts that state into another state.
  - – defines a relation of accessibility among states.
  - – represents the conditions of applicability of a state and corresponding transformation function

State Space

A State space is the set of all states reachable from the initial state.

Definitions of terms:

- ✓ A state space forms a graph (or map) in which the nodes are states and the arcs between nodes are actions.
- ✓ In state space, a path is a sequence of states connected by a sequence of actions.
- ✓ The solution of a problem is part of the map formed by the state space.

Structure of State Space

The Structures of state space are trees and graphs.

- ✓ Tree has only one path to a given node; i.e., a tree has one and only one path from any point to any other point.
- ✓ Graph consists of a set of nodes (vertices) and a set of edges (arcs).
- ✓ Arcs establish relationships (connections) between the nodes; i.e., a graph has several paths to a given node.

Search process explores the state space.

- ✓ In the worst case, the search explores all possible paths between the initial state and the goal state.

Problem solution

- In the state space, a solution is a path from the initial state to a goal state or sometime just a goal state.
- A solution cost function assigns a numeric cost to each path; It also gives the cost of applying the operators to the states.
- A solution quality is measured by the path cost function
- An optimal solution has the lowest path cost among all solutions.

Problem Solving

Problem Definitions :

- ✓ A problem is defined by its elements and their relations.
- ✓ To provide a formal description of a problem, we need to do following:

a. Define a state space that contains all the possible configurations of the relevant objects, including some impossible ones.

b. Specify one or more states, that describe possible situations, from which the problem-solving process may start. These states are called initial states.

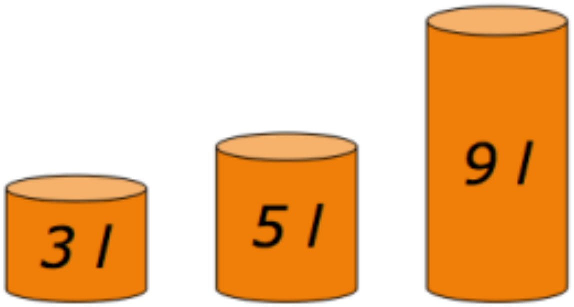
c. Specify one or more states that would be acceptable solution to the problem. These states are called goal states.

d. Specify a set of rules that describe the actions (operators) available.

Problem Solving

- ✓ The problem can then be solved by using the rules, in combination with an appropriate control strategy, to move through the problem space until a path from an initial state to a goal state is found.
- ✓ This process is known as search.
    - Search is fundamental to the problem-solving process.
    - Search Is a general mechanism that can be used when more direct method is not known.
    - Search provides the framework into which more direct methods for solving sub-parts of a problem can be embedded.
- ✓ **A very large number of AI problems are formulated as search problems.**

Example: Measuring Problem



Problem : Using these buckets, measure 7 liters of water

Example: Measuring Problem

Measure 7 liters of water using a 3-liter, a 5-liter, and a 9-liter buckets.

- ✓ Formulate goal: Have 7 liters of water in 9-liter bucket
- ✓ Formulate problem:
  - States: amount of water in the buckets
  - Operators: Fill bucket from source, empty bucket
- ✓ Find solution: sequence of operators that bring you from current state to the goal state

Example: Measuring Problem

Solution 1:

a	b	c	
0	0	0	start
3	0	0	
0	0	3	
3	0	3	
0	0	6	
3	0	6	
0	3	6	
3	3	6	
1	5	6	
0	5	**7**	**goal**

- **Solution 2:**

a	b	c	
0	0	0	start
0	5	0	
3	2	0	
3	0	2	
3	5	2	
3	**0**	**7**	**goal**

Searching Process

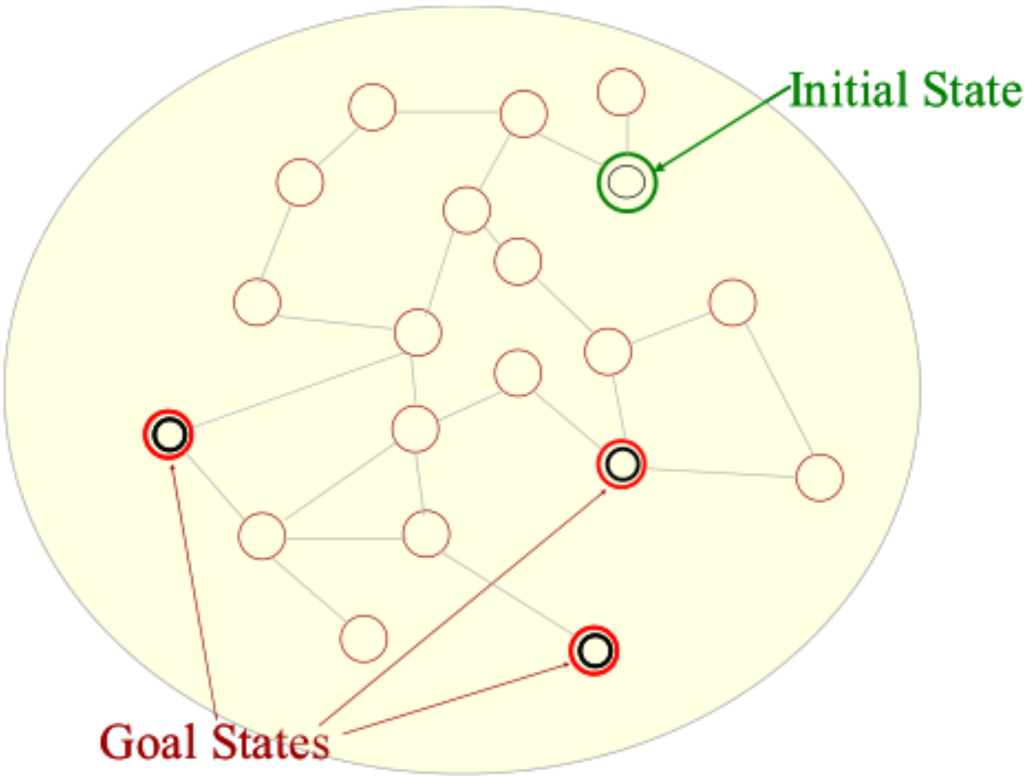
The generic searching process can be very simply described in terms of the following steps:

Do until a solution is found or the state space is exhausted.

1. Check the current state
2. Execute allowable actions to find the successor states.
3. Pick one of the new states.
4. Check if the new state is a solution state

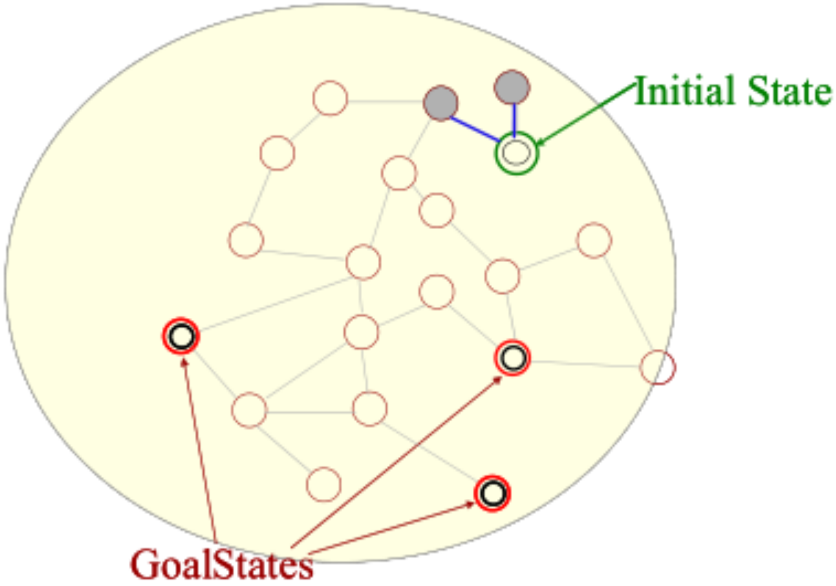
If it is not, the new state becomes the current state and the process is repeated

Illustration of Searching Process



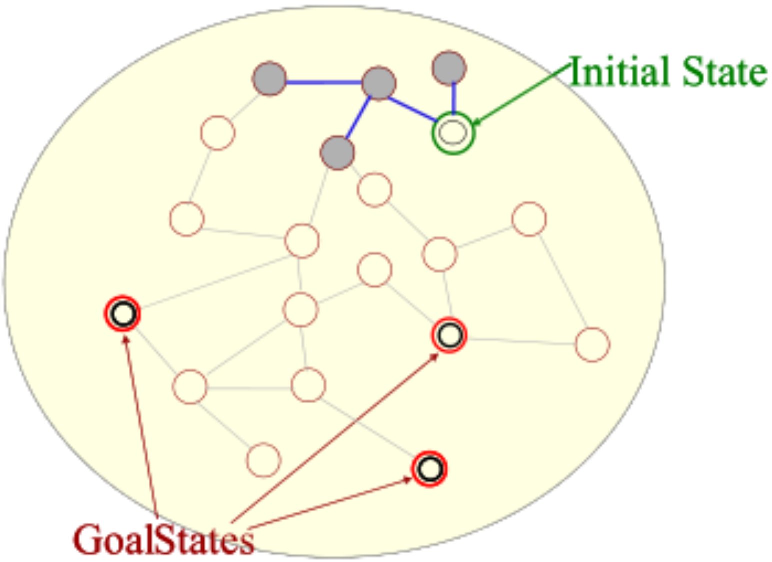
s_0 is the initial state.
The successor states are the adjacent states in the graph.
There are three goal states.

Illustration of Searching Process



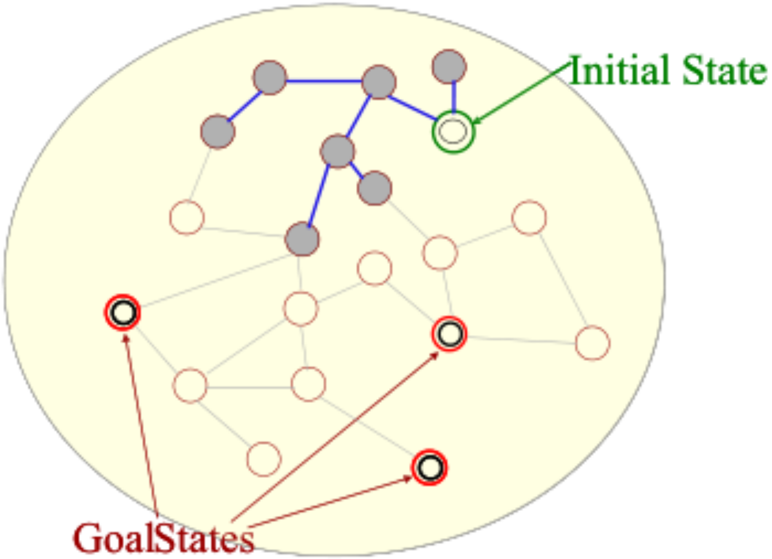
The two successor states of the initial state are generated.

Illustration of Searching Process



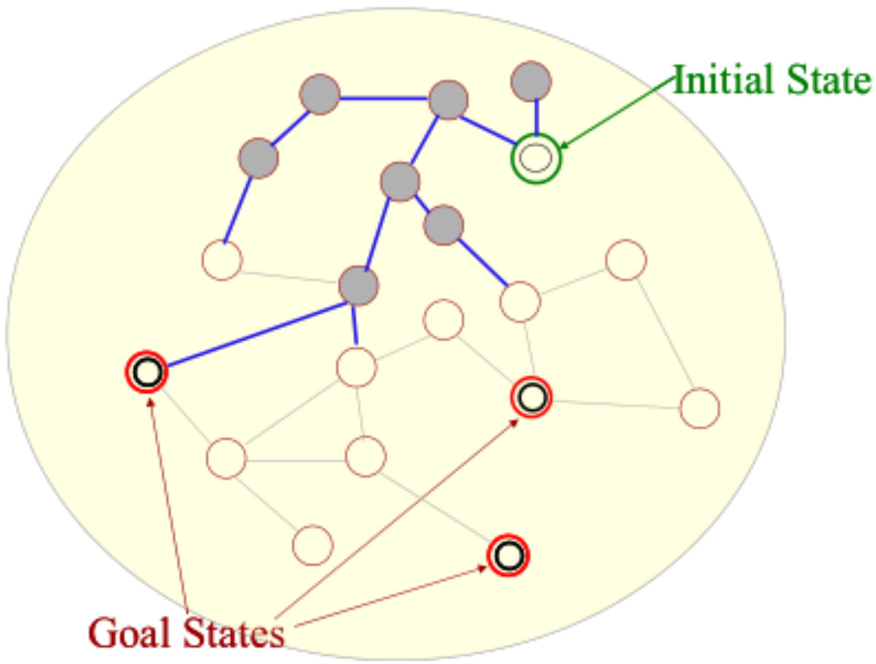
The successors of these states are picked and their successors are generated.

Illustration of Searching Process



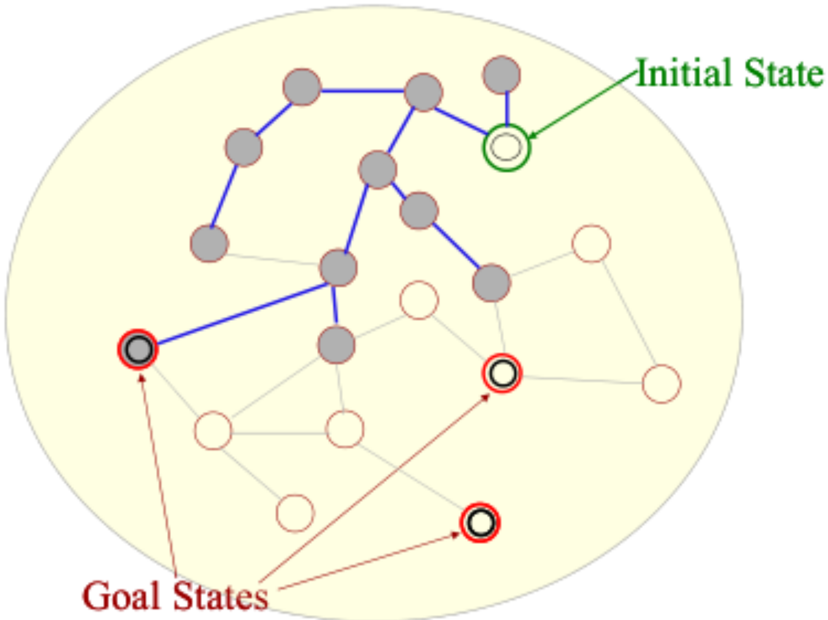
Successors of all these states are generated.

Illustration of Searching Process



The successors are generated.

Illustration of Searching Process



A goal state has been found.

Well-defined problems

A problem can be defined formally by five components:

1. The initial state that the agent starts in.
2. A description of the possible actions available to the agent.
3. A description of what each action does
4. The goal test, which determines whether a given state is a goal state.
5. A path cost function that assigns a numeric cost to each path.

Example – 8 Puzzle

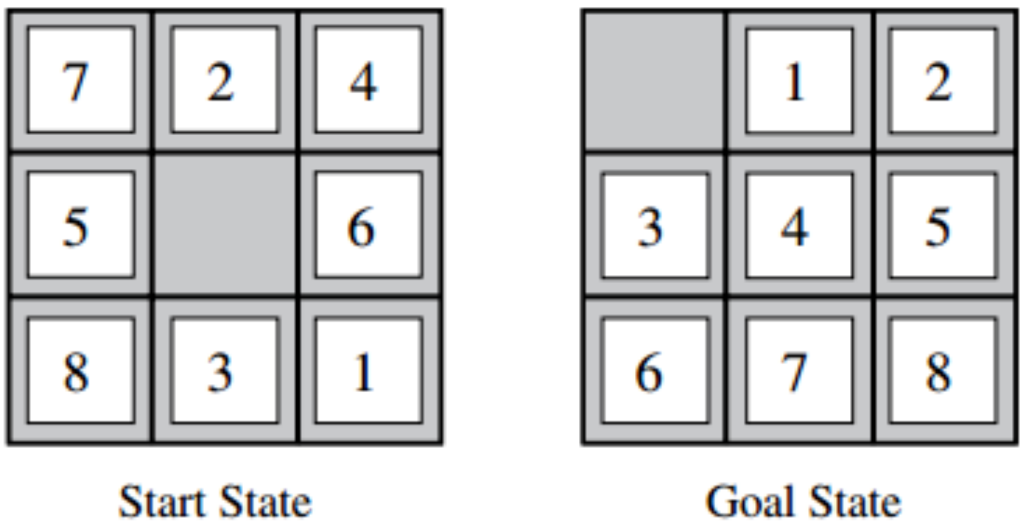
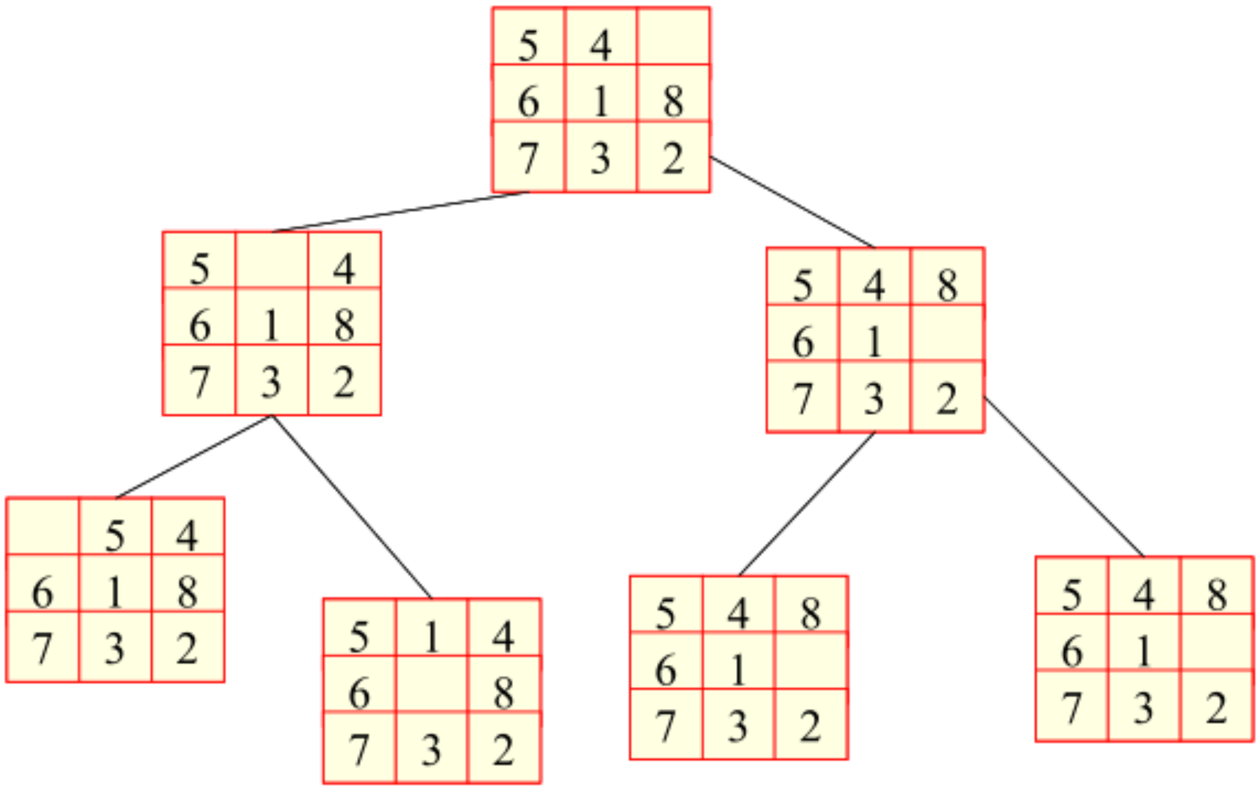


Figure 3.4 A typical instance of the 8-puzzle.

Example – 8 Puzzle

- **States:** A state description specifies the location of each of the eight tiles and the blank in one of the nine squares.
- **Initial state:** Any state can be designated as the initial state.
- **Actions:** The simplest formulation defines the actions as movements of the blank space Left, Right, Up, or Down. Different subsets of these are possible depending on where the blank is.
- **Transition model:** Given a state and action, this returns the resulting state; for example, if we apply Left to the start state in Figure 3.4, the resulting state has the 5 and the blank Switched.
- **Goal test:** This checks whether the state matches the goal configuration shown in Figure 3.4. (Other goal configurations are possible.)
- **Path cost:** Each step costs 1, so the path cost is the number of steps in the path.

Example – 8 Puzzle



8-puzzle partial state space

Example – 8-Queens Problem

- ✓ The goal of the 8-queens problem is to place eight queens on a chessboard such that no queen attacks any other.
- ✓ (A queen attacks any piece in the same row, column or diagonal.)

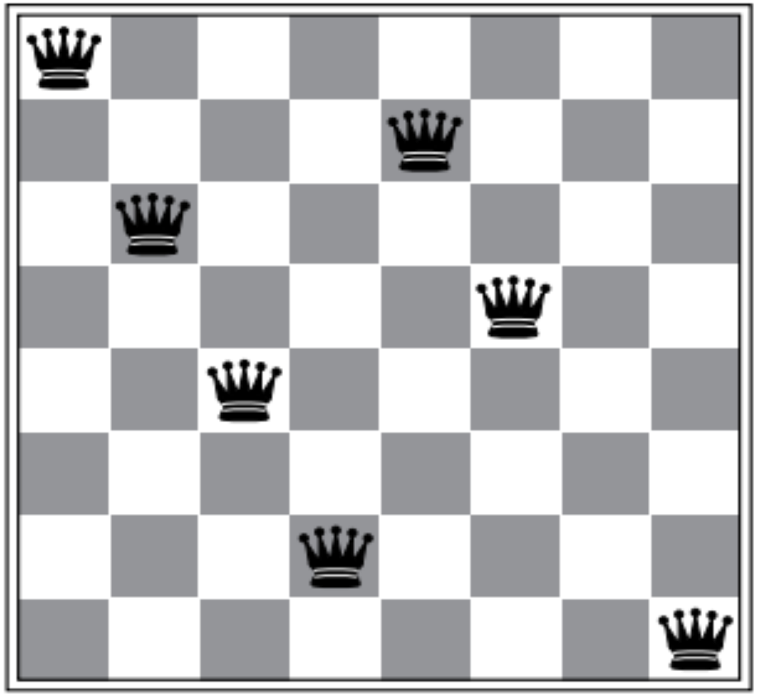


Figure 3.5 Almost a solution to the 8-queens problem. (Solution is left as an exercise.)

Example – 8-Queens Problem

- ✓ States: Any arrangement of 0 to 8 queens on the board is a state.
- ✓ Initial state: No queens on the board.
- ✓ Actions: Add a queen to any empty square.
- ✓ Transition model: Returns the board with a queen added to the specified square.
- ✓ Goal test: 8 queens are on the board, none attacked.